METHODS, SYSTEMS AND COMPUTER PROGRAM PRODUCTS FOR RESUMING SNA APPLICATION-CLIENT COMMUNICATIONS AFTER LOSS OF AN IP NETWORK CONNECTION

FIELD OF THE INVENTION

The present invention relates generally to SNA communications sessions and, more particularly, to recovery of such communications sessions after loss of a network connection.

BACKGROUND OF THE INVENTION

Mainframe computer systems are widely used for a variety of data processing functions. For example, many corporate financial and accounting systems were developed for, and operate on, mainframe computing systems. Although the use of personal computers and personal computer networks has proliferated in recent years, mainframe computer systems, referred to as legacy systems, are expected to remain popular for many types of data processing for years to come.

A mainframe computer system typically includes multiple user terminals connected to a mainframe host computer. Various software applications may be resident on, or accessible to, the host computer. The user terminals, which are sometimes referred to as client terminals or "clients", communicate with the mainframe host computer via a host communications

system. The International Business Machine Corporation's ("IBM") 3270 terminal system is one of the more widely-used systems for communicating with host computers.

Typically, a 3270 terminal does not have its own data processing capability. Instead, a 3270 terminal may display screens generated by applications running on a host, and may communicate information from a user back to a host for processing. A user interface of an IBM 3270 terminal system comprises a plurality of screens that are generated by host applications for display on the 3270 terminal. Over the past few years, many 3270 terminals have been replaced with personal computers (PCs) configured to communicate with the mainframe computer system via 3270 terminal emulation software.

In many applications, 3270 terminals, or other data processing devices (e.*g.*, a PC) emulating a 3270 terminal, now access the host computer via the Internet. For example, in systems operating under the TN3270E protocols, the 3270 terminals may access application software on a host computer via a combination of a Transmission Control Protocol/Internet Protocol (TCP/IP) protocol connection between the TN3270E client terminal and a TN3270E server, and via a Systems Network Architecture (SNA) session between the TN3270E server and the SNA application software on the host mainframe computer. Such SNA application software or "SNA applications" refer to software applications designed for use by a plurality of terminals interconnected by an SNA network or a variant of an SNA network.

Figure 1 depicts such a mainframe computing system environment in which several TN3270E client terminals

20 communicate with several SNA applications **30** via a TN3270E server **40**. The SNA applications **30** may be resident on one or more mainframe computers or other data processing systems (not shown in **Figure 1**), or, alternatively, may be physically located separate from the one or more main frame computers/data processing devices but accessible by such devices. As shown in **Figure 1**, the TN3270E clients **20** are connected to the TN3270E server **40** via Internet protocol or "IP" connections **25**. The TN3270E server **40** is connected to the SNA applications **30** via SNA sessions **35**.

In the TN3270E environment of **Figure 1**, the SNA sessions extend only between the TN3270E server **40** and the SNA applications **30**. Thus, the TN3270E server **40** converts IP communications from one of the TN3270E clients **20** into SNA format and forwards them to one of the SNA applications **30**. The TN3270E server **40** likewise converts SNA communications received from an SNA application **30** into IP format and forwards them to the TN3270E client over the IP network.

In many legacy systems, each 3270 terminal was assigned a unique logical unit or "LU" name. Typically, legacy systems originated as hard-wired systems, and the LU name served to identify the particular terminal/user that was accessing the mainframe host computer. Consequently, many SNA software applications that were developed to run on these legacy systems included software routines that made decisions based on the LU name of the device/user accessing the terminal. Consistent with the requirements of these SNA applications, under the TN3270E protocol, a data processing device that is acting as a TN3270E client may specify an LU name when

connecting to a TN3270E server. Typically, the TN3270E server will have information, such as Logmode information, regarding one or more LU names which may attempt to connect to SNA applications through the TN3270E server. Upon receiving a connection request from a TN3270E client, the TN3270E server may provide the SNA application certain of the information regarding the TN3270E client before establishing the SNA session. Moreover, typically a TN3270E server will only allow one SNA session at a time involving a particular LU name.

SUMMARY OF THE INVENTION

Systems, methods and computer program products are provided for preserving a session between an SNA application and a TN3270E server after loss of an Internet protocol ("IP") network connection between the TN3270E server and a TN3270E client that is communicating with the SNA application via the SNA session. In embodiments of the invention, the IP connection between the TN3270E server and the TN3270E client is reestablished and then the TN3270E server may forward a screen refresh request to the SNA application. The TN3270E server may also receive a screen refresh from the SNA application, and may forward this screen refresh to the TN3270E client over the reestablished IP connection. This screen refresh may comprise the last data screen that was forwarded from the SNA application which was acknowledged as received by the TN3270E client. The IP connection may be a TCP/IP connection or some other type or layer connection in an IP network.

One way in which the TN3270E server may forward the screen refresh request to the SNA application is by

sending an LUSTAT message to the SNA application. When such an LUSTAT message is used, a user logon screen may be received from the SNA application in response to the LUSTAT message, which, may be forwarded to the TN3270E client. Logon information may be received from the TN3270E client in response to the logon screen, and the authenticity of the received logon information checked. The screen refresh may then be forwarded to the TN3270E client over the reestablished IP connection if the received logon information is authentic. The SNA application may also send the screen refresh to the TN3270E server, for forwarding to the TN3270E client, without first sending the logon screen to the TN3270E client (via the TN3270E server) and receiving logon information in response thereto.

In further embodiments of the present invention, a TCP/IP connection between a TN3270E client and a TN3270E server is reestablished after loss of a first TCP/IP connection between the TN3270E client and the TN3270E server. In such embodiments, the TN3270E server may receive a connection request, where the connection request specifies an LU name that was specified in the first TCP/IP connection. In response to receiving this request, the TN3270E server may transmit a query addressed to the TN3270E client over the first TCP/IP connection. If a response to the query is not received within a specified time period, the TN3270E server establishes a second TCP/IP connection in response to the connection request, and resumes communications over this second TCP/IP connection. In these embodiments, the query may comprise a query to which the TN3270E client automatically responds such as a timemark request. Furthermore, the identity of the TN3270E client may be

authenticated prior to resuming communications with the TN3270E client over the second TCP/IP connection.

As will be appreciated by those of skill in the art in light of the present disclosure, embodiments of the present invention may include methods, systems (devices) and/or computer program products.

BRIEF DESCRIPTION OF THE DRAWINGS

Figure 1 schematically illustrates host communications between several SNA applications and several TN3270E client terminals over an IP network.

Figure 2 illustrates a TN3270E communications environment incorporating embodiments of the present invention.

Figure 3 is a block diagram of portions of the communications server for the OS/390 operating system incorporating embodiments of the present invention.

Figure 4 is a flow chart illustrating operations according to embodiments of the present invention.

Figure 5 is a flow chart illustrating operations according to alternative embodiments of the present invention.

Figure 6 is a flow chart illustrating operations according to additional embodiments of the present invention.

DETAILED DESCRIPTION OF PREFERRED EMBODIMENTS

The present invention now will be described more fully hereinafter with reference to the accompanying drawings, in which preferred embodiments of the invention are shown. This invention may, however, be embodied in many different forms and should not be construed as limited to the embodiments set forth herein; rather, these embodiments are provided so that

this disclosure will be thorough and complete, and will fully convey the scope of the invention to those skilled in the art.

As will be appreciated by one of skill in the art, the present invention may be embodied as a method, data processing system, and/or computer program product. Accordingly, the present invention may take the form of an entirely hardware embodiment, an entirely software embodiment or an embodiment combining software and hardware aspects. Furthermore, the present invention may take the form of a computer program product on a computer-usable storage medium having computer-usable program code means embodied in the medium. Any suitable computer readable medium may be utilized including hard disks, CD-ROMs, optical storage devices, a transmission media such as those supporting the Internet or an intranet, or magnetic storage devices.

Computer program code for carrying out operations of the present invention may be written in an object oriented programming language such as Java®, Smalltalk or C++. However, the computer program code for carrying out operations of the present invention may also be written in conventional procedural programming languages, such as the "C" programming language. The program code may execute entirely on a single computer, or be distributed so as to execute on a plurality of computers and/or data processing devices.

The present invention is described below with reference to flowchart illustrations and/or block diagrams of methods, apparatus (systems) and computer program products according to embodiments of the invention. It will be understood that each block of the flowchart illustrations and/or block diagrams, and combinations of blocks in the flowchart illustrations

and/or block diagrams, can be implemented by computer program instructions. These computer program instructions may be provided to a processor of a general purpose computer, special purpose computer, or other programmable data processing apparatus to produce a machine, such that the instructions, which execute via the processor of the computer or other programmable data processing apparatus, create means for implementing the functions specified in the flowchart and/or block diagram block or blocks.

These computer program instructions may also be stored in a computer-readable memory that can direct a computer or other programmable data processing apparatus to function in a particular manner, such that the instructions stored in the computer-readable memory produce an article of manufacture including instruction means which implement the function specified in the flowchart and/or block diagram block or blocks.

The computer program instructions may also be loaded onto a computer or other programmable data processing apparatus to cause a series of operational steps to be performed on the computer or other programmable apparatus to produce a computer implemented process such that the instructions which execute on the computer or other programmable apparatus provide steps for implementing the functions specified in the flowchart and/or block diagram block or blocks.

Pursuant to embodiments of the present invention, methods, systems and computer program products are provided for preserving a session between an SNA application and a TN3270E server after loss of an IP network connection between the TN3270E server and a TN3270E client that is communicating with the SNA application via the SNA session. **Figure 2** depicts a

TN3270E computing environment incorporating embodiments of the present invention. As shown in **Figure 2**, a TN3270E client terminal **60** accesses an SNA application **64** that is resident on a remote mainframe computer via a TN3270E server **62**. The TN3270E client **60** communicates with the TN3270E server **62** over a TCP/IP transport layer connection **58** in an Internet protocol network **56**. The TN3270E server **62** converts communications from the TN3270E client **60** into a format suitable for transmission over an SNA network, and then forwards these communications to SNA application **64** via an SNA session **54** in SNA network **50**. The TN3270E server **62** likewise converts SNA communications from SNA application **64** into a format suitable for transmission over TCP/IP connection **58** to TN3270E client **60**.

In a conventional TN3270E environment, if the TCP/IP connection **58** depicted in **Figure 2** is broken (*i.e.*, due to a switch or router failure), the SNA session **54** would have to be restarted once the TCP/IP connection **58** was reestablished (or an alternate TCP/IP connection implemented in its place). Loss of this SNA session could result in several negative impacts, including (i) increased use of resources in the SNA network **50** at the SNA nodes along the session path resulting from clean-up of the original SNA session, (ii) increased use of the resources in the SNA session **54**, (iii) potential confusion which may result from SNA session failure messages that may result when the original SNA session **54** is torn down and (iv) loss of SNA network availability due to the delay associated with tearing down and reestablishing SNA session **54**.

In embodiments of the present invention, the operations for preserving the SNA session across the

reconnect of a TCP/IP connection may be carried out by the TN3270E server **62**. However, it will be appreciated in light of the teachings of the present disclosure that the operations could be carried out by a separate data processing unit. One type of TN3270E server **62** which could be modified in accordance with the teachings of the present invention is the communications server for the OS/390 operating system ("CS-OS/390"). **Figure 3** depicts such a CS-OS/390 operating system **70**.

As illustrated in **Figure 3**, the CS-OS/390 operating system **70** includes a TN3270E server **60** that acts as an interface between the IP network **56** and the SNA network **50**. The CS-OS/390 operating system **70** communicates with the IP network **56** via an IP stack **72**, which is operatively coupled to the TN3270E server **60**. The CS-OS/390 operating system **70** likewise includes an SNA stack **74** (such as the Virtual Telecommunications Access Manager or "VTAM") via which the operating system may communicate with devices in the SNA network **50**. The SNA stack **74** is likewise operatively coupled to the TN3270E server **60**. Profiles **76** associated with client terminals in the IP network may be stored by the TN3270E server **60**.

Pursuant to embodiments of the present invention, methods, systems and computer program products are provided for expediting the reconnect of a TN3270E client **60** to a TN3270E server **62** after loss of a connection, such as the TCP/IP connection **58** in the IP network **56** (see **Figure 2**). In certain situations, the TN3270E client **60** may become aware of the loss of the TCP/IP connection **58** to the TN3270E server **62** before the TN3270E server **62** becomes aware that the TCP/IP

connection **58** has been lost. This may occur, for example, when the break in the TCP/IP connection **58** happens at, or very close to, the TN3270E client **60**, and/or because the TN3270E server **62** has a relatively long value (*e.g.*, on the order of minutes) on its inactivity timer.

In these situations, the TN3270E client **60** may attempt to reestablish the TCP/IP connection **58** with the TN3270E server **62** before the TN3270E server **62** becomes aware that the original TCP/IP connection **58** is lost. However, since the TN3270E server **62** does not realize that the original TCP/IP connection **58** has been lost, the TN3270E server **62** may interpret the connection request from the TN3270E client that seeks to reestablish the original TCP/IP connection **58** as a second TN3270E client **65** attempting to connect using an already allocated LU name (as the connection request would specify the LU name specified in the connection request that established the original TCP/IP connection **58**).

In conventional systems (see **Figure 1**), when the TN3270E server **40** receives a connection request specifying an existing LU name, the TN3270E server **40** typically will reject the connection request as an improper attempt to use an already allocated LU name. The TN3270E server **40** may thereafter continue to reject additional reconnection requests from the original TN3270E client 20 until the inactivity period expires and the TN3270E server **40** realizes that the original TCP/IP connection **25** has been lost.

Pursuant to the teachings of the present invention, the delays which may occur in the above-described scenario may be reduced by sending a query to

the original TN3270E client **60** in the event that the TN3270E server **62** receives a second connection request specifying the LU name associated with the original TCP/IP connection **58**. **Figure 4** depicts operations associated with such queries according to embodiments of the present invention.

As shown in **Figure 4**, operations may begin upon the receipt of a connection request which specifies an LU name that is already associated with the existing (first) TCP/IP connection **58** between the TN3270E server **62** and the first TN3270E client **60** (block **100**). Upon receiving this connection request, a query is transmitted over the first TCP/IP connection **58** that is addressed to the first TN3270E client **60** (block **102**). Upon transmitting the query, the TN3270E server **62** may set a response timer (block **104**). If a response to the query is received before the response timer times out (block **106**), the connection request is rejected (block **108**). If, on the other hand, the response timer times out (block **106**) without receipt of any response from the first TN3270E client **60**, the TN3270E server **62** assumes that the first TCP/IP connection **58** has been lost, and thus may accept the connection request (block **110**) and establish a new TCP/IP connection (block **112**). Once this new TCP/IP connection is established, the TN3270E server **62** may resume communications with the TN3270E client **60** over this new TCP/IP connection (block **114**).

Pursuant to other embodiments of the present invention, methods, systems and computer program products are provided for preserving the SNA session **54** between the SNA application **64** and the TN3270E server **62** during the time required to reconnect the TN3270E

client **60** that is accessing the SNA application **64** to the TN3270E server **62** after loss of the IP network connection **58** (see **Figure 2**). **Figure 5** illustrates operations according to embodiments of the present invention whereby the SNA session **54** between the SNA application **64** and the TN3270E server **62** may be preserved across the reconnect of the lost IP connection **58** between the TN3270E server **62** and the TN3270E client **60**.

Conventionally, as illustrated in **Figure 1**, upon the loss of an IP connection **25**, the SNA session **35** is torn down and reestablished anew, as the TN3270E server **40** and/or SNA application **30** typically do not know how much data was received by the TN3270E client **20** prior to loss of the IP network connection **25**. As shown in **Figure 5**, according to embodiments of the present invention, operations begin upon the loss of the IP connection **58** between the TN3270E server **62** and the TN3270E client **60** that is associated with the SNA session **54** (block **120**). Upon identifying the loss of the IP connection **58**, the TN3270E client **60** and/or TN3270E server **62** act to reestablish the IP connection **58** (block **122**). This may be accomplished, for example, using the operations of **Figure 4** or via conventional steps for reestablishing such a connection. Once the IP connection **58** is reestablished, the TN3270E server **62** may forward a request to the SNA application **64** for a screen refresh (block **124**). It will be appreciated, however, that the TN3270E client **60** or another device monitoring the session could likewise forward the screen refresh request to the SNA application **64**. A screen refresh request constitutes a request that the application retransmit the data required to redisplay a

screen that was previously displayed on the client terminal. So long as the SNA application **64** includes such a screen refresh capability, it may retransmit to the TN3270E client **60** the data corresponding to the last screen (which herein may be referred to as a "screen") which was confirmed as having been received by the TN3270E client **60** (or, alternatively, may send the next screen or a prior screen) (block **126**). As these operations provide a mechanism for resynchronizing the TN3270E client **60** and the SNA application **64** after the loss of the IP connection **58**, there is no need for the TN3270E server **62** to terminate the original SNA session **54** after loss of the IP connection **58**. Consequently, the SNA session **54** is preserved across the IP connection reconnect, and the SNA application **64** is resynchronized with the TN3270E client **60** according to the operations of **Figure 5**.

Figure 6 depicts operations according to further embodiments of the present invention which illustrate an exemplary way in which the IP reconnection procedures of **Figure 4** could be combined with the operations for preserving an SNA session across the reconnect of **Figure 5**. The operations of **Figure 6** further include optional client authentication procedures which may be used to provide a more secure reconnect of the lost IP connection.

As shown in **Figure 6**, operations commence when the TN3270E server receives a connection request (block **150**). As noted above, such a connection request typically specifies an LU name. The TN3270E server **62** determines whether or not the specified LU name has already been allocated to an existing IP connection between the TN3270E server **62** and a first TN3270E client

60 (block **152**). If the LU name is not currently allocated, the connection request is processed normally (block **154**), and operations are concluded.

If the LU name specified in the connection request is currently allocated to an existing IP connection (block **152**), the TN3270E server **62** may proceed to authenticate the client that sent the connection request (block **156**). The TN3270E server **62** may use any of a variety of criteria to determine whether or not to perform such authentication (*e.g.*, authenticate all connection requests, authenticate none, authenticate connection requests associated with certain LU names, authenticate connection requests received over certain network paths, etc.).

If authentication is to be performed, the TN3270E server **62** performs the authentication procedures (block **158**) to determine if the connection request is authentic. Such authentication may be done, for example, in a CS-OS/390 based TN3270E server by having the Resource Access Control Facility ("RACF") of the OS/390 perform a full Secure Socket Layer ("SSL") client authentication on a digital X.509 certificate received from the TN3270E client along with the connection request. Such an X.509 certificate may be "burnt" onto the hard drive at the TN3270E client terminal or input via a "swipe" card, and may include, among other things, a user identification. The user at the TN3270E client **60** may also be required to input a password which is forwarded to the TN3270E server **62** along with the X.509 certificate, typically in an encrypted format. A connection request may be considered "authentic" if the password received from the client is the correct password for the received user identification.

If the client authentication procedures indicate

that the connection request is not authentic (block **160**), then the connection request is rejected (block **162**), and operations are concluded.

If, on the other hand, at block **160** it is determined that the connection request is authentic, the certificate forwarded along with the connection request is compared to the certificate associated with the TN3270E client **60** to which the specified LU name has already been allocated (block **164**). If the two certificates are not the same (block **166**), then the connection request may be rejected (block **162**). If instead, at block **166** the two certificates are the same, the TN3270E server **62** sends a message to the first TN3270E client **60** over the original IP connection **58** which requires a response from the TN3270E client **60** (block **170**). Typically, this request will be for a response that may be immediately provided by the TN3270E client **60** without the need for user input, such as a timemark request (*i.e.*, a request that the terminal send the value of its time-of-day clock).

Upon sending this message, the TN3270E server **62** starts a response timer (block **172**) and monitors for a response to the message from the TN3270E client **60**. If a response is received before the response timer times out (block **174**), the connection request is rejected (block **162**). If a response has not been received by the time the response timer times out (block **174**), the connection request is allowed and the requested IP connection is established (block **176**).

Once the new IP connection has been established, the TN3270E server **62** may send an LUSTAT message to the SNA application **64** (block **178**). An LUSTAT message refers to a command flow which requests that the SNA

application resend or "refresh" the last screen which the TN3270E client **60** acknowledged as having received correctly. Thus, by sending the LUSTAT message to the SNA application **64**, the TN3270E server **62** can prompt the SNA application **64** to resynchronize the TN3270E client **60** after the lost IP connection **58** has been restored, and do so in a manner that preserves the SNA session **54** between the SNA application **64** and the TN3270E server **62**.

It will be appreciated that the operations depicted in the flow charts of **Figures 4-6** need not always be performed in the particular order indicated. For instance, in **Figure 6**, the authentication operations of blocks **158, 160** might be carried out after the timemark was forwarded to the TN3270E client **60** at block **170**. Likewise, typically the operations of blocks **170** (sending the timemark) and **172** (starting the response timer) are carried out substantially simultaneously, but they could also be carried out sequentially, in either order. It will be appreciated that various other modifications could be made to the order in which the operations are performed without departing from the scope or teachings of the present invention.

As noted above, in embodiments of the present invention, a response timer is set when a query is sent from the TN3270E server **62** to the TN3270E client **60** after the TN3270E server **62** receives a connection request that specifies an already allocated LU name. This response timer may be set to a fixed value or may be a variable parameter that is dependent, for example, on the LU name allocated, the network path of the first IP connection, network delay data or various other parameters. Typically, the response timer is set to be somewhat greater than the expected response time for a

response to a query, yet less than the setting on the inactivity timer at the TN3270E server **62**. In this manner, the response timer may run out before the inactivity timer at the TN3270E server **62**, thereby alerting the TN3270E server **62** to the loss of the connection faster than normal, facilitating a faster reconnect of the lost IP connection.

As is also noted above, the operations for speeding up the reconnect of a TCP/IP or other connection in the IP network may include an authentication procedure. Such authentication may be advantageous because depending upon the setting on the response timer, it may be possible that the response timer will run out even though the original TCP/IP connection has not been lost due to delays in the network in either sending the timemark or receiving the TN3270E client's response thereto. By including authentication measures, it is possible to guard against unauthorized access to the SNA session by another user that sends a connection request specifying an already allocated LU name. The robustness of the security scheme used may depend upon the sensitivity of the information involved, the settings on the response timer, and/or various other parameters.

In embodiments of the present invention, the SNA session between the TN3270E server **62** and the SNA application **64** is preserved by resynchronizing the TN3270E client **60** and the SNA application **64** using a screen refresh procedure. In this manner, the TN3270E server **62** and/or the SNA application **64** may supply the necessary session information to the TN3270E client **60** after loss of the IP connection **58** (since the client typically retains no knowledge of the SNA session **54** across the reconnect).

As noted above, in embodiments of the present

invention, the TN3270E server **62** requests that the SNA application **64** perform a screen refresh by sending an LUSTAT message to the SNA application **64**. In response to the LUSTAT message, the SNA application **64** typically sends a logon screen to the TN3270E client **60** that prompts the end user to enter a user identification and password. Once validated (where validation may be performed by either the TN3270E server **62** or the SNA application **64**), the last acknowledged data screen sent prior to the reconnect is retransmitted to the TN3270E client **60**.

As will be appreciated by those of skill in the art, some SNA applications may not include a screen refresh capability, as they may not buffer the last screen which was acknowledged as received by the TN3270E client. However, even with these applications the methods, systems and computer program products of the present invention may improve network performance as these applications typically still prompt the TN3270E client to re-logon to the SNA application via the log-on screen. Such a re-logon may not be viewed by the network as the termination of the original SNA session and, consequently, the original session is not terminated and reestablished. Additionally, some SNA applications may not support receipt of an LUSTAT message. With respect to these applications, loss of the IP connection **25** will typically result in termination of the session.

Embodiments of the present invention are described and claimed herein with reference to TN3270E clients and servers. As used herein, the term "TN3270E client" is meant to refer to any 3270 terminal, or any device emulating a 3270 terminal, in which the device or its user specifies an LU name or other identifying feature

when establishing a network connection and/or when accessing an SNA application. Accordingly, it will be appreciated that the term "TN3270E client" is meant to not only encompass client terminals and devices operating in the TN3270E environment, but also client terminals and devices operating in modified versions of that environment so long as they specify an LU name or other identifying feature. The term "TN3270E server" is likewise intended to encompass both servers operating in the TN3270E environment and servers operating in modified versions or variations of the TN3270E environment. Furthermore, as described above, the term "TN3270E server" refers to any system which interfaces between the IP connection and the SNA session. Thus, the TN3270E server may appear as a server to the TN3270E client and as a terminal to the SNA application.

Reference is also made herein to "SNA sessions." It will be appreciated that such SNA sessions refer to a session layer connection in the SNA network architecture or its equivalent in variations or modified versions of the layered SNA network architecture.

Additionally, reference is made herein to clients and servers. As used herein, a "client" refers to a computer, workstation, terminal or other data processing device that accesses over a network an application, data set, etc. from another data processing/storage device. A "server" refers to a data processing system which provides a response to a communication from a client. It will be appreciated that the same data processing device may be both a client and server with respect to different operations.

The foregoing is illustrative of the present invention and is not to be construed as limiting thereof. Although a few exemplary embodiments of this

invention have been described, those skilled in the art will readily appreciate that many modifications are possible in the exemplary embodiments without materially departing from the novel teachings and advantages of this invention. Accordingly, all such modifications are intended to be included within the scope of this invention as defined in the claims. In the claims, means-plus-function clauses are intended to cover the structures described herein as performing the recited function and not only structural equivalents but also equivalent structures. Therefore, it is to be understood that the foregoing is illustrative of the present invention and is not to be construed as limited to the specific embodiments disclosed, and that modifications to the disclosed embodiments, as well as other embodiments, are intended to be included within the scope of the appended claims. The invention is defined by the following claims, with equivalents of the claims to be included therein.